

Autonomous Learning for Face Recognition in the Wild via Ambient Wireless Cues

Chris Xiaoxuan Lu*
University of Oxford
xiaoxuan.lu@cs.ox.ac.uk

Xuan Kan*
Tongji University
kanxuan1996@gmail.com

Bowen Du
University of Warwick
B.Du@warwick.ac.uk

Changhao Chen
University of Oxford
changhao.chen@cs.ox.ac.uk

Hongkai Wen
University of Warwick
hongkai.wen@dcs.warwick.ac.uk

Andrew Markham
University of Oxford
andrew.markham@cs.ox.ac.uk

Niki Trigoni
University of Oxford
niki.trigoni@cs.ox.ac.uk

John A. Stankovic
University of Virginia
jas9f@virginia.edu

ABSTRACT

Facial recognition is a key enabling component for emerging Internet of Things (IoT) services such as smart homes or responsive offices. Through the use of deep neural networks, facial recognition has achieved excellent performance. However, this is only possible when trained with hundreds of images of each user in different viewing and lighting conditions. Clearly, this level of effort in enrolment and labelling is impossible for wide-spread deployment and adoption. Inspired by the fact that most people carry smart wireless devices with them, e.g. smartphones, we propose to use this wireless identifier as a supervisory label. This allows us to curate a dataset of facial images that are unique to a certain domain e.g. a set of people in a particular office. This custom corpus can then be used to finetune existing pre-trained models e.g. FaceNet. However, due to the vagaries of wireless propagation in buildings, the supervisory labels are noisy and weak. We propose a novel technique, AutoTune, which learns and refines the association between a face and wireless identifier over time, by increasing the inter-cluster separation and minimizing the intra-cluster distance. Through extensive experiments with multiple users on two sites, we demonstrate the ability of AutoTune to design an environment-specific, continually evolving facial recognition system with entirely no user effort.

CCS CONCEPTS

• **Human-centered computing** → *Ubiquitous and mobile computing*; • **Information systems** → *Information retrieval*; • **Computing methodologies** → *Lifelong machine learning*.

KEYWORDS

Adaptation of Learning Systems; Cross-modality Association; Face Recognition; Internet of Things

*Both authors contributed equally to the paper.

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

© 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-6674-8/19/05.

<https://doi.org/10.1145/3308558.3313398>

ACM Reference Format:

Chris Xiaoxuan Lu, Xuan Kan, Bowen Du, Changhao Chen, Hongkai Wen, Andrew Markham, Niki Trigoni, and John A. Stankovic. 2019. Autonomous Learning for Face Recognition in the Wild via Ambient Wireless Cues. In *Proceedings of the 2019 World Wide Web Conference (WWW'19)*, May 13–17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, Article 0, 11 pages. <https://doi.org/10.1145/3308558.3313398>

1 INTRODUCTION

Facial recognition and verification are key components of smart spaces, e.g., offices and buildings for determining who is where. Knowing this information allows a building management system to tailor ambient conditions to particular users, perform automated security (e.g., opening doors for the correct users without the need for a swipe card), and customize smart services (e.g., coffee dispensing). A vast amount of research over the past decades has gone into designing tailored systems for facial recognition and with the advent of deep learning, progress has accelerated. As an example of a state-of-the-art face recognizer, FaceNet achieves extremely high accuracies (e.g., 99.5%) on very challenging datasets through the use of a low dimensional embedding, allowing similar faces to be clustered through their Euclidean distance [14, 26]. However, when directed transferred to operate in ‘in the wild’, subject to variable lighting conditions, viewing angle and appearance changes, performance of off-the-shelf pre-trained classifiers degrades significantly, with accuracies around 15% not being uncommon. The solution to this is to obtain a large, labelled corpus of data for a particular environment, with hundreds of annotated images per user. Given access to such a hypothetical dataset, it is then possible to fine-tune the pre-trained classifier on out-domain data to adapt to the new environment and achieve excellent performance.

However, the cost of labelling and updating the corpus (e.g. to enrol new users) is prohibitive for most critical applications and therefore, will naturally limit the use and uptake of facial recognition as a ubiquitous technology in emerging Internet of Things (IoT) applications. On the other side, people often, but not always, carry smart devices (phones, fitness devices etc). Wang et al. [36] advocated that although these devices do not provide fine-grained enough positioning capability to act as a proxy for presence, they can be used to indicate that a user *might* be present in an area

with a co-located camera. In this work, we take a step forward and further utilize device presence as weak supervision signals for the purposes of fine-tuning a classifier. The goal now becomes how to take an arbitrary, pre-trained recognition network and tune it from a generic classifier to a highly specific classifier, optimized for a certain environment and group of people. We note that the aim is to make the network better and better at this specific goal, but it would likely perform poorly if transferred directly to a different environment. This is the antithesis of the conventional view of generalized machine learning, but is ideally suited for the problem of environment specific facial recognition, as opposed to generic facial recognition.

The technical challenge is that there is not a 1:1 mapping between a face and a wireless identity, rather we need to solve the association between a set of faces and a set of identities over many sessions or occasions. To further complicate the problem, the sets are not pure i.e. the set of faces can contain additional faces from people not of interest (e.g. visitors). Equally well, due to the vagaries of wireless transmission, the set of wireless identifiers will contain additional identifiers e.g. from people in the next office. Furthermore, it is also possible to have missing observations e.g. because a person was not facing the camera or because someone left their phone at home.

In this work, we present AutoTune, a system which can be used to gradually improve the performance of facial recognition systems in the wild, with zero user effort, tailoring them to the visual specifics of a particular smart space. We demonstrate state-of-the-art performance in real-world facial recognition through a number of experiments and trials.

In particular, our contributions are:

- We observe and prove that wireless signals of users’ devices provide valuable, albeit noisy, clues for face recognition. Namely, wireless signals can serve as a weak label. Such weak labels can replace the human annotated face images in the wild to save intensive effort.
- We create AutoTune, a novel pipeline to simultaneously label face images in the wild and adapt the pre-trained deep neural network to recognize the faces of users in new environments. The key idea is to repeat the face-identity association and network update in tandem. To cope with observation noise, we propose a novel probabilistic framework in AutoTune and design a new stochastic center loss to enhance the robustness of network fine-tuning.
- We deployed AutoTune in two real-world environments and experimental results demonstrate that AutoTune is able to achieve $> 0.85 F_1$ score of image labeling in both environments, outperforming the best competing approach by $> 25\%$. Compared to the best competing approach, using the features extracted from the fine-tuned model and training a classifier based on the cross-modality labeled images can give a $\sim 19\%$ performance gain for online face recognition.

The rest of this paper is organized as follows. §2 introduces the background of this work. System overview is given in §3. We describe the AutoTune solution in §4 and §5. System implementation details are given in §6. The proposed approach is evaluated and compared with state of the art methods in §7. Finally, we discuss and outlook future work in §8 and conclude in §9.

2 RELATED WORK

Deep face recognition: Face recognition is arguably one of the most active research areas in the past few years, with a vast corpus of face verification and recognition work [23, 31, 40]. With the advent of deep learning, progress has accelerated significantly. Here we briefly overview state-of-the-art work in Deep Face Recognition (DFR). Taigman et al. pioneered this research area and proposed DeepFace [30]. It uses CNNs supervised by softmax loss, which essentially solves a multi-class classification problem. When introduced, DeepFace achieved the best performance on the Labeled Face in the Wild (LFW) [9] benchmark. Since then, many DFR systems have been proposed. In a series of papers [27, 28] Sun et al. extended on DeepFace incrementally and steadily increased the recognition performance. A critical point in DFR happened in 2015, when researchers from Google [26] used a massive dataset of 200 million face identities and 800 million image face pairs to train a CNN called Facenet, which largely outperformed prior art on the LFW benchmark when introduced. A point of difference is in their use of a “triplet-based” loss [5], that guides the network to learn both inter-class dispersion and inner-class compactness. Recently proposed RTFace [36] not only achieves high recognition accuracy but also operates at the full frame rates of videos. Although the above methods have proven remarkably effective in face recognition, the training needs a vast amount of labeled images to train the supervised DFR network. A large amount of labeled data is not always achievable in a particular domain, and using a small amount of training data will incur poor generalization ability in the wild.

Cross-modality Matching: Cross-modal matching has received considerable attention in different research areas. Methods have been developed to establish mappings from images [7, 11, 34] and videos [33] to textual descriptions (e.g., captioning), developing image representation from sounds [20, 21], recognizing speaker identities from Google calendar information [17], and generating visual models from text [41]. In cross-modality matching between images and radio signals, however, work is very limited and all dedicated to trajectory tracking of humans [1, 22, 32]. The field of face recognition via wireless signals is an unexplored area.

3 AUTOTUNE OVERVIEW

3.1 System Model

We consider a face recognition problem with m people of interest (POI) and each subject owns one WiFi-enabled device, e.g., a smartphone. We denote the identity set as $\mathcal{Y} = \{y_j | j = 1, 2, \dots, m\}$. The set of observed POI’s devices in a particular environment is denoted by $\mathcal{L} = \{l_j | j = 1, 2, \dots, m\}$, e.g., a set of MAC addresses. We assume the mapping from device MAC addresses \mathcal{L} to the user identity \mathcal{I} is known. A collection of face images $\mathcal{X} = \{x_j | j = 1, 2, \dots, n\}$ is cropped from the surveillance videos in the same environment. Note that, to mimic the real-world complexity, our collection includes the faces of both POI and some non-POI, e.g., subjects with unknown device MAC addresses. We then assign face and device observations to different subsets based on their belonging events $\mathcal{E} = \{e_j | j = 1, 2, \dots, h\}$. An event e_j is the setting in which people interact with each other in a specific part of the environment for a given time interval. It is uniquely identified by three attributes: effective timeslot, location, and participants. Fig. 1 demonstrates a

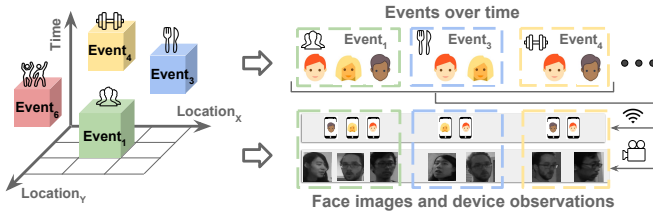


Figure 1: An illustrative example of *events* and showing their linked *observations*. An *event* is uniquely determined by spatial-temporal attributes and its participants. An event is also linked with two types of observations. Face observation are images cropped from surveillance videos and device observation are sniffed MAC address of participants’ devices.

few examples of events. Lastly, we also have a deep face representation model f_θ pre-trained on public datasets that contains no POI. Such model is trained with metric losses, e.g., triplet loss so that the learned features could bear a good property for clustering [26].

In this sense, the problem addressed by AutoTune is assigning IDs to images from noisy observations of images and WiFi MAC addresses, and using such learned ID-image associations to tune the pre-trained deep face representation model automatically.

3.2 System Architecture

AutoTune is based on two key observations: i) although collected by different modalities, both face images and device MAC addresses are linked with the identities of users who attend certain events; and ii) the tasks of model tuning and face-identity association should not be dealt with separately, but rather progress in tandem. Based on the above insights, AutoTune works as follows (see Fig 2):

- **Heterogeneous Data Sensing.** This module collects facial and WiFi data (device attendance observations) through surveillance cameras and WiFi sniffers¹ in a target environment. Given the face images and sniffed WiFi MAC addresses, AutoTune first segments them into events based on the time and location they were captured.
- **Cross-modality Labeling.** This module first clusters face images based on their appearance similarity computed by the face representation model, and also taking into account information on device attendance in events. Each image cluster should broadly correspond to a user, and the cluster’s images are drawn from a set of events. We then assign each cluster to the user whose device has been detected in as similar as possible set of events.
- **Model Updates.** Once images are labeled with user identity labels, this module then fine-tunes the pre-trained face representation model. AutoTune further uses the cluster labels to update our belief on which device (MAC address) has participated in each event.

The sensing module is one-off and we will detail its implementation in §6.1. Labeling and model update modules are iteratively repeated until the changes in the user attendance model become

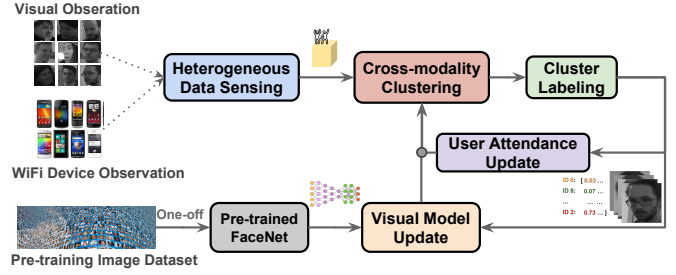


Figure 2: Workflow of AutoTune. AutoTune consists of 5 steps i) Heterogeneous Data Sensing ii) Cross-Modality Clustering iii) Cluster Labeling iv) Visual Model Update v) User Attendance Update. AutoTune sequentially repeats the above five steps until the changes in the user attendance are negligible.

negligible. The tuned model derived in the last iteration is regarded as the one best adapted to POI recognition in the new environment.

4 CROSS-MODALITY LABELING

In this section, we introduce the labeling module in AutoTune. The challenge in this module is that collected facial images and sniffed WiFi data are temporally unaligned. For example, detecting a device WiFi address does not imply that the device owner will be captured by the camera at the exact instant and vice versa. Such mismatches in cross-modality data distinguish our problem from prior sensor fusion problems, where both multiple sensors are observing a temporal evolving system. In order to tackle the above challenge, we leverage the diverse attendance patterns in events and use a two-step procedure in isolated. Images \mathcal{X} are firstly grouped together into clusters across all sessions, and we then associate clusters with device IDs (i.e., labels) \mathcal{L} based on their similarity in terms of event attendance.

4.1 Cross-modality Clustering

Heterogeneous Features. Given a pre-trained face representation model f_θ , an image x_i can be translated to the feature vector z_i . Unlike conventional clustering that merely depends on the face feature similarity, AutoTune merges face images *across events* into a cluster (potentially belonging to the same subject) by incorporating attendance information as well. Recall that device attendance already reveals the identities of subjects (in the form of MAC addresses) in a particular event, and the captured images in the same event may contain the faces of these subjects as well. Despite the noise in observations, the overlapped subjects in different events can be employed as a prior that guides the image clustering. For example, if there are no shared MAC addresses sniffed in two events, then it is very likely that the face images captured in these two events should lie in different clusters. Formally, for an event e_i , we denote a device attendance vector as $\mathbf{u}_k = (u_k^1, u_k^2, \dots, u_k^m)$, where $u_k^j = 1$ if device l_j is detected in event e_k . In this way, we could construct a heterogeneous feature $\tilde{z}_i = [z_i, \mathbf{u}_k]$ for an image x_i collected in the event e_k . Note that, as all images captured in a same event have the same attendance vector, this device attendance part are essentially enforced on cross-event image comparison.

¹<https://www.wireshark.org/>

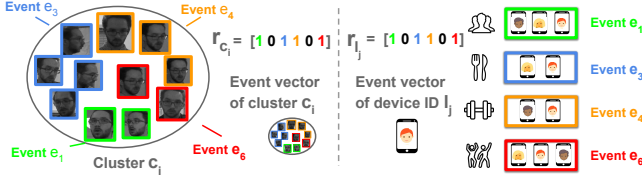


Figure 3: Event vectors of an image cluster c_i and device ID l_j . The k th element of the vector $r_{c_i}^k$ is set to 1 only if it contains images attached with the event e_k . The event vector r_{l_j} of the device ID l_j is similarly developed. This insight leads to Eq. 1.

Face Image Similarity. Given an image x_i captured in event e_k (i.e., $\tilde{z}_i = [z_i, \mathbf{u}_k]$) and an image x_j captured in event e_p (i.e., $\tilde{z}_j = [z_j, \mathbf{u}_p]$), the likelihood that two cross-event face images belong to the same subject is conditioned on two factors: i) the similarity of their feature representation between z_i and z_j ; and ii) the overlap ratio between the attendance observations in their corresponding events \mathbf{u}_k and \mathbf{u}_p respectively. The resulted joint similarity is a log likelihood $\log(\Pr(x_i = x_j))$ defined as follows:

$$\Pr(x_i = x_j) \propto \frac{\exp(\beta * |\mathbf{u}_k \otimes \mathbf{u}_p|)}{\exp(\beta * |\mathbf{u}_k \oplus \mathbf{u}_p|)} * \exp(-D(z_i, z_j))$$

$$\log(\Pr(x_i = x_j)) \propto \underbrace{\beta * (|\mathbf{u}_k \otimes \mathbf{u}_p| - |\mathbf{u}_k \oplus \mathbf{u}_p|)}_{\text{attendance similarity}} - \underbrace{D(z_i, z_j)}_{\text{feature distance}}$$

Here \otimes and \oplus are element-wise AND and OR, and $|\cdot|$ here is the L^1 -norm. z is the features transformed by the face representation model and D is a distance measure between face features. β , analogous to the regularization parameter in composite loss functions, is a hyper-parameter that controls the contributions of the attendance assistance and feature similarity. The above derivation is inspired by the Jaccard coefficients, with the difference lying in the log function. The rationale behind the term $|\mathbf{u}_k \oplus \mathbf{u}_p|$ is that the more different subjects attending events, the more uncertain that any two images drawn across these events will point to the same subject. In contrast, when the intersection $|\mathbf{u}_k \otimes \mathbf{u}_p|$ is significant enough, the chance that these two images point to the same subject will become higher. This joint similarity can also be explained from a Bayesian perspective. The attendance similarity of two events can serve as a prior that two cross-event images belong to the same subject and the feature similarity can be seen as the likelihood. Together they determine the posterior probability that two cross-event images fall into the same cluster. Based on the above joint similarity, images across events are grouped into clusters $C = \{c_i | i = 1, 2, \dots, g\}$. We will soon discuss how to determine the number of clusters g based on the complete set of MAC addresses \mathcal{L} in the next section.

4.2 Cluster Labeling

Fine-grained ID Association. After clustering, an image cluster is linked with multiple events that are associated with its member images. Naturally, we introduce an event vector $\mathbf{r}_{c_i} = (r_{c_i}^1, r_{c_i}^2, \dots, r_{c_i}^h)$ for an image cluster c_i , where h is the total number of events. $r_{c_i}^k$ is set to 1 only if c_i contains images from event e_k . Fig. 3 shows an example of how an event vector is developed. Similarly, for a

device ID l_j , its corresponding event vector $\mathbf{r}_{l_j} = (r_{l_j}^1, r_{l_j}^2, \dots, r_{l_j}^h)$ can be determined by inspecting its occurrences in all WiFi sniffing observations. $r_{l_j}^k$ is set to 1 only if the device ID (MAC address) l_j is detected in the event e_k . The intuition behind ID association is that a device and a face image cluster of the same subject should share the most consistent attendance pattern in events, reflected by the similarity of their event vectors. Based on this intuition, AutoTune assigns clusters with device IDs based on the matching level of their event vectors. Formally, a matching problem of bipartite graph can be formulated as follows:

$$\mathcal{L}_A = \sum_{ij} a_{ij} (\mathbf{r}_{c_i} - \mathbf{r}_{l_j})^2$$

$$\sum_{1 \leq j \leq m} a_{ij} = 1, \quad \forall i \in \{1, \dots, g\} \quad (1)$$

where the solution of the binary variable a_{ij} assigns a device ID to a cluster. We note that when $m < g$, AutoTune adds dummy nodes to create the complete bipartite graph. Then the complete bipartite graph problem can be solved by the Hungarian algorithm [10].

Probabilistic Labeling via Soft Voting. We now obtain the association between face images and device IDs. However, in practice, the number of clusters g can vary due to the captured face images outside the POI, e.g., face images of short-term visitors. The choice of g has a significant impact on the performance of clustering [6] which could further affect the following fine-tuning performance. To cope with this issue, AutoTune sets the number of clusters g greater than the number of POI m . With a larger number of clusters, although there are some unused clusters of non-POI after clustering, the association step will sieve them and only choose the most consistent m clusters with the event logs of m device IDs. Lastly, for every image, its assigned ID is finalized by decision voting on the individual association results computed with different g , which will be soon introduced in the next paragraph. Typically, a majority voting procedure would give a hard ID label for an image and use these $\langle \text{Image}, \text{ID} \rangle$ pairs as training data to update the face representation model. However, due to the noise in device presence observations and errors in clustering, there will be mis-labeling in the training data that may confuse the model update. To account for the uncertainty of ID assignment, we adopt soft labels for voting. Then instead of voting for the most likely ID y_i for an image x_i , we introduce a probability vector $\mathbf{y}_i = (y_{i,1}, y_{i,2}, \dots, y_{i,m})$. Specifically, every image is associated with all POI and its associated soft label is derived by the votes for the subjects divided by the total number of votes. In this way, the soft label of each image is a valid probability distribution that sums up to 1. For instance, $y_{i,j} = 0.4$ means that there are 40% associations assigning the ID j to the image i . Moreover, the soft labeled data $\langle x_i, \mathbf{y}_i \rangle$ is compatible with the computation of cross-entropy.

Number of Clusters. We are now in a position to describe how to select the number of clusters g . We observed that, with a proper clustering algorithm (i.e., agglomerative clustering in our choice), non-POI's images will form separate small clusters but they will not be chosen in our follow-up association step. This is because that the number of devices ID m is always less than the number of clusters ($g \in [2 * m, 5 * m]$), and only those images clusters with the most consistent event attendances are associated. The small clusters of

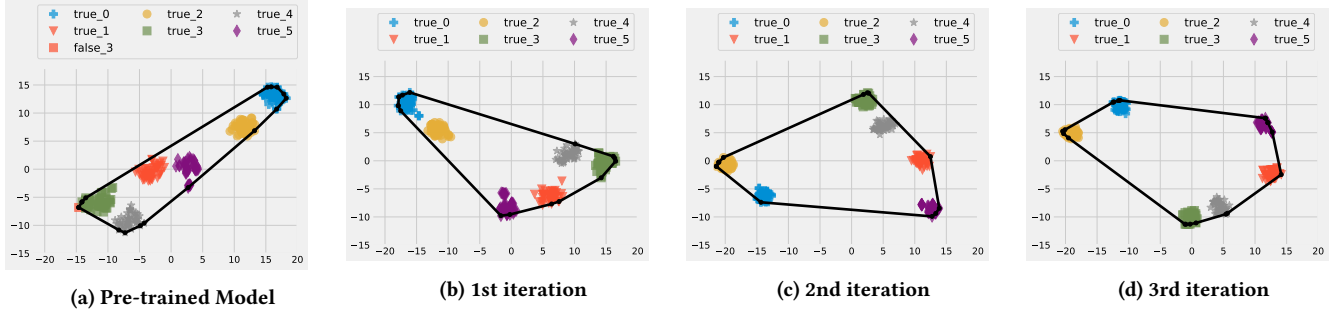


Figure 4: An example of visual model update (§5.1) in AutoTune with a six-subject subset. Note that the pre-trained model in Fig. 4a has a narrow convex hull (black lines), clusters are dispersed and there is mis-classification (see false_3 in Fig. 4a) as the pre-trained features are not discriminative enough to differentiate unseen subjects in a new environments. Over time, clusters become more compact and move further away from each other, leading to purer clustering results.

outlier images often have scattered attendance vector and will be ignored in association. Therefore, soft voting with varying number of clusters can reinforce the core images of POI by giving large probability and assigns non-POI’s images with small probability.

5 MODEL UPDATES

We are now in a position to introduce the model update in AutoTune. At every iteration, AutoTune updates the last-iteration face representation model f_{θ}^{τ} to $f_{\theta}^{\tau+1}$, by taking the labeled images as inputs. To correct the device observation errors, AutoTune leverages these label images to update device attendance vector to \mathbf{u}^{τ} for all events.

5.1 Visual Model Update

Discriminative Face Representation Learning. Face representation learning optimizes a representation loss \mathcal{L}_R to enforce the learnt features as discriminative as possible. Strong discrimination bears two properties: inter-class dispersion and intra-class compactness. Inter-class dispersion pushes face images of different subjects away from one another and the intra-class compactness pulls the face images of the same subject together. Both of the properties are critical to face recognition. At iteration τ , given the current labels y_i^{τ} for the i th face image x_i and the transformed features $\mathbf{z}_i^{\tau} = f_{\theta}^{\tau}(x_i)$, the representation loss \mathcal{L}_R is determined by a composition of softmax loss and center loss:

$$\begin{aligned} \mathcal{L}_R &= \mathcal{L}_{softmax} + \mathcal{L}_{center} \\ &= \underbrace{\sum_i -\log\left(\frac{e^{W_{y_i}^T \mathbf{z}_i^{\tau} + b_{y_i}^{\tau}}}{\sum_{j=1}^m e^{W_{l_j}^T \mathbf{z}_i^{\tau} + b_{l_j}^{\tau}}}\right)}_{\text{softmax loss}} + \underbrace{\sum_i \frac{\lambda}{2} \|\mathbf{z}_i^{\tau} - \mathbf{o}_{y_i^{\tau}}\|^2}_{\text{center loss}} \end{aligned} \quad (2)$$

where W and b are the weights and bias parameters in the last fully connected layer of the pre-trained model. $\mathbf{o}_{y_i^{\tau}}$ denotes a centroid feature vectors by averaging all feature vectors with the same identity label y_i . The center loss \mathcal{L}_{center} explicitly enhances the intra-class compactness while the inter-class dispersion is implicitly strengthened by the softmax loss $\mathcal{L}_{softmax}$ [37]. λ is a hyper-parameter that balances the above sub-losses.

Stochastic Center Loss The center loss \mathcal{L}_{center} in Eq. 2 is shown to be helpful to enhance the intra-class compactness [37]. However,

we cannot directly adopt it for fine-tuning as computing the centers requires explicit labels (see Eq. 2) of images, while the association steps above only provide probabilistic ones through soft labels. To solve this, we propose a new loss called *stochastic center loss* \mathcal{L}_{stoc} to replace the center loss. Similar to the idea of fuzzy sets [38], we allow each face image to belong to more than one subject. The membership grades indicate the degree to which an image belongs to each subject and can be directly retrieved from the soft labels and the stochastic center \mathbf{o}_k^{τ} for the k -th identity is given as:

$$\mathbf{o}_k^{\tau} = \frac{\sum_i^n \mathbf{z}_i^{\tau} * y_{i,k}^{\tau}}{\sum_i^n y_{i,k}^{\tau}} \quad (3)$$

This gives the stochastic center loss as follows:

$$\mathcal{L}_{stoc} = \sum_i \sum_k y_{i,k}^{\tau} * \|\mathbf{z}_i^{\tau} - \mathbf{o}_k^{\tau}\|^2 \quad (4)$$

We leave the softmax loss $\mathcal{L}_{softmax}$ the same as in Eq. 2, because the soft labels are compatible with the computation of cross-entropy. The new representation loss to minimize is:

$$\mathcal{L}_R = \mathcal{L}_{softmax} + \mathcal{L}_{stoc} \quad (5)$$

AutoTune updates the model parameters θ^{τ} to $\theta^{\tau+1}$ based on the gradients of $\nabla_{\theta} \mathcal{L}_R$, which are calculated via back propagation of errors. Compared with the dataset used for pre-training, which is usually in the order of millions [4], the data used for fine-tuning is much smaller (several thousands). The mis-match between the small training data and the complex model architecture could result in overfitting. To prevent overfitting, we use the dropout mechanism in training, which is widely adopted to avoid overfitting [13]. Meanwhile, as observed in [8, 25], the soft label itself can play the role of a regularizer, and make the trained model robust to noise and reduce overfitting. Fig. 4 illustrates the effect of model update.

5.2 User Attendance Update

The device presence observations by WiFi sniffing are noisy because the WiFi signal of a device is opportunistic and people do not carry/use their devices all the time. Based on the results of the cluster labelling step introduced in §4.2, we have the opportunity to update our belief on which users attended each event. The update

ALGORITHM 1: AutoTune

Input: pre-trained model f_θ^1 , images \mathcal{X} , POI's device IDs \mathcal{L} , threshold ξ
Output: adapted model f_θ^* , corrected attendance observations I^* , soft image labels Y^*

Initialize: Given sniffed \mathcal{L} in all events \mathcal{E} , compute attendance vector \mathbf{u}^0
 $\tau = 1$

```
while  $\sqrt{\frac{1}{h} \sum_{k=1}^h \|\mathbf{u}_k^{(\tau)} - \mathbf{u}_k^{(\tau-1)}\|^2} > \xi$  do
   $\mathcal{Z}^\tau = f_\theta^\tau(\mathcal{X})$  ▷ feature transformation
  for  $g \leftarrow 2 * m$  to  $5 * m$  do
     $C^\tau \leftarrow \text{cross\_modality\_clustering}(\mathcal{Z}^\tau, \mathbf{u}^\tau, g)$  ▷ §4.1;
     $A_g^\tau \leftarrow \text{cluster\_labeling}(C^\tau, \mathbf{r}_c^\tau, \mathbf{r}_l^\tau)$  ▷ §4.2;
  end
   $\mathcal{Y}^\tau \leftarrow \text{soft\_voting}(A^\tau)$  ▷ §4.2;
   $\mathbf{o}^\tau \leftarrow \text{stochastic\_center}(\mathcal{Y}^\tau, \mathcal{Z}^\tau)$  ▷ §5.1;
   $f_\theta^{\tau+1} \leftarrow \text{visual\_model\_update}(\mathcal{X}, \mathcal{Y}^\tau, \mathbf{o}^\tau, f_\theta^\tau)$  ▷ §5.1;
   $\mathbf{u}^{\tau+1} \leftarrow \text{user\_attendance\_update}(\mathcal{Y}^\tau, \mathbf{u}^\tau)$  ▷ §5.2;
   $\tau \leftarrow \tau + 1$ 
end
```

mechanism is as follows: Each image is associated with a user probability vector, whose elements denote the probability that the image correspondences to a particular user. By averaging the user probability vectors of all images that have been drawn from the same event e_k , and normalizing the result, we can estimate the user attendance of this event. The elements of the resulting user attendance vector $\hat{\mathbf{u}}_k^\tau$ denote the probabilities of different users attending event e_k .

We can now use $\hat{\mathbf{u}}_k^\tau$ as a correction to update our previous wifi attendance vector \mathbf{u}_k^τ as follows:

$$\mathbf{u}_k^{\tau+1} = \mathbf{u}_k^\tau - \gamma \cdot (\mathbf{u}_k^\tau - \hat{\mathbf{u}}_k^\tau) \quad (6)$$

where γ is a pre-defined parameter that controls the ID update rate. In principle, a large update rate will speed up the convergence rate, at the risk of missing the optima. AutoTune sequentially repeats the above steps of clustering, labelling and model updates, until the changes ξ in the user attendance model are negligible (≤ 0.01 in our case). Algorithm. 1 summarizes the workflow.

6 IMPLEMENTATION

In this section, we introduce the implementation details of AutoTune (code available at <https://github.com/Wayfear/Autotune>).

6.1 Heterogeneous Data Sensing

Face Extraction. This module consists of a front-end remote camera and a back-end computation server². Specifically, the remote cameras in our experiment are diverse and include *GoPro Hero*⁴, *Mi Smart Camera*⁴ and *Raspberry Pi Camera*⁵. We modified these cameras so that they are able to communicate and transfer data to the back-end through a wireless network. To avoid capturing excess data without people in it, we consider a motion-triggered mechanism with a circular buffer. It works by continuously taking low-resolution images, and comparing them to one another for

changes caused by something moving in the camera's field of view. When a change is detected, the camera takes a higher-resolution video for 5 seconds and reverts to low resolution capturing. All the collected videos are sent to the backend at every midnight. On the backend, a cascaded convolutional network based face detection module [39] is used to drop videos with no face in them. The cropped faces from the remaining videos are supplied to AutoTune. **WiFi Sniffing.** This module is realized on a WiFi-enabled laptop running Ubuntu 14.04. Our sniffer uses Aircrack-ng⁶ and tshark⁷ to opportunistically capture the WiFi packets in the vicinity. The captured packet has unencrypted information such as transmission time, source MAC address and the Received Signal Strengths (RSS). As AutoTune aims to label face images for POI, our WiFi sniffer only records the packets containing MAC addresses of POI's and discards them otherwise, so as to not harvest addresses from people who have not given consent. A channel hop mechanism is used in the sniffing module to cope with cases where the POI's device(s) may connect to different WiFi networks, namely, on different wireless channels. The channel hop mechanism forces the sniffing channel to change by every second and monitor the active channels periodically (1 second) in the environment. The RSS value in the packet implies how far away the sniffed device is from the sniffer [18, 19]. By putting the sniffer near the camera, we can use a threshold to filter out those devices with low RSS values, e.g., less than -55 dBm in this work, as they are empirically unlikely to be within the camera's field of view.

Event Segmentation. Depending on the context, the duration of events can be variable. However, for simplicity we use fix-duration events in this work. Specifically, we split a day into 12 intervals, each of which is 2 hours long. We then discard those events that have neither face images nor sniffed MAC addresses of POI.

6.2 Face Recognition Model

The face recognition model used in AutoTune is the state-of-the-art FaceNet [26].

Pre-training. FaceNet adopts the Inception-ResNet-v1 [29] as its backbone and its weights are pre-trained on the VGGFace2 dataset [4]. This dataset contains 3.31 million images of 9131 subjects, with an average of 362.6 images for each subject. Images are downloaded from Google Image Search and have large variations in pose, age, illumination, ethnicity and profession. Pre-training is supervised by the triplet loss [5] and the training protocols, e.g. parameter settings, can be found in [26]. We found that the learnt face representation by FaceNet is generalizable and it is able to achieve an accuracy of 99.65% on the LFW face verification task⁸. Note that, FaceNet [26] not only learns a powerful recognition model but gives a very discriminative feature representation that can be used for clustering.

Fine-tuning by AutoTune. The fine-tuning process has been explained in §5.1, and here we provide some key implementation details. After each round of label association (see §4.2), the labeled data is split into a training set and validation set, with a ratio of 8 : 2 respectively. The pre-trained FaceNet is then fine-tuned on the training set, and the model that achieves the best performance

²The study has received ethical approval R50950

³<https://shop.gopro.com/cameras>

⁴<https://www.mi.com/us/mj-panorama-camera/>

⁵<https://www.raspberrypi.org/products/camera-module-v2/>

⁶<https://www.aircrack-ng.org/>

⁷<https://www.wireshark.org/docs/man-pages/tshark.html>

⁸<http://vis-www.cs.umass.edu/lfw/>

on the validation set is saved. Note that the fine-tuning process in AutoTune does not involve the test set. The online testing is performed on a held-out set that is collected on different days. To enhance the generalization ability, we use dropout training for regularization [35]. The dropout ratio is set to 0.2. We set the batch size to 50 and one fine-tuning takes 100 epochs.

6.3 System Configuration

Face Detection. As discussed in §6.1, we use a cascaded convolution network to detect faces in videos. It is cascaded by three sub-networks, a proposal network, a refine network and an output network. Each of them can output bounding boxes of potential faces and the corresponding detection probabilities, i.e., confidences. Face detection with small confidence will be discarded early in the process and not sent to the next sub-network. In this work, the confidence threshold is set to 0.7, 0.7 and 0.9 for three sub-networks respectively. Following the original setting in [39], we set the minimal face size in detection to 40×40 pixels.

Setup of Face Clustering. In §4.1, we use a clustering algorithm to merge the images across events. Specifically, the clustering algorithm used in AutoTune is agglomerative clustering [2], which is a method that recursively merges the pair of clusters that minimally increases a given linkage distance. The similarity metric adopted here is Euclidean distance. A linkage criterion determines which distance to use between sets of data points. In AutoTune, this linkage criterion is set to the average distances of all sample pairs from two sets. As introduced in Sec. 4.2, the number of clusters g is determined by the number of POI m . We vary g from $2 * m$ to $5 * m$ and proceed soft voting, to account for the extra clusters of non-POI’s face images and ambiguous/outlier images of POI.

7 EVALUATION

In this section, we evaluate the AutoTune extensively on datasets collected from both real-world experiments and simulation. We deployed two testbeds, one in the UK and the other in China, and collected datasets as described in the previous section. The simulation dataset is developed based on a public face datasets.

7.1 Evaluation Protocols

Competing Approaches. We compare the performance of AutoTune with the 3 competing approaches:

- **Template Matching (TM)** [3] employs a template matching method to assign ID labels to clusters of face images. This is the most straightforward method that is used when one or more profile photos of POI are available, e.g., crawled from their personal homepage or Facebook.
- **One-off Association (OA)** [16] uses one-off associations to directly label the image clusters without fine-tuning of the face representation model itself.
- **Deterministic AutoTune (D-AutoTune)** is the deterministic version of AutoTune. In D-AutoTune, the association and update steps are the same as in AutoTune, but it adopts hard labels rather than soft labels and uses the simple center loss instead of the proposed stochastic center loss (see §5.1).

Evaluation Metrics. AutoTune contains two main components, offline label assignment and online inference. For the offline label

assignment, we evaluate its performance with the following metrics: TP, TN, FP, FN are true positive, true negative, false positive and false negative respectively. Each metric captures different aspects of classification [24]. Online face recognition has two kinds of tests, face identification and verification. We follow [14, 15] to use Cumulative Match Characteristic (CMC) for the evaluation of online face identification.

7.2 Offline Cross-modality Face Labeling

AutoTune automatically labels images captured in the wild by exploiting their correlations with the device presences. The quality of image labeling is crucial for the follow-up face recognition in the online stage. In this section, we investigate the image labeling performance of AutoTune.

7.2.1 Data Collection. We deployed AutoTune at the testbeds in two countries, with different challenging aspects.

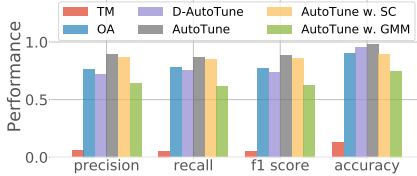
UK Site: This first dataset is collected in a commercial building in the UK. We deploy the heterogeneous sensing front-ends, including surveillance cameras and WiFi sniffers, on a floor with three different types of rooms: office, meeting room and kitchen. 24 long-term occupants work inside can freely transit across these rooms. These occupants are naturally chosen as people of interest (POI). For the office, face images are captured with a surveillance camera that faces the entrance. The presence logs of occupants’ WiFi MAC addresses are collected by a sniffer that is situated in the center of the room for the same time period. Besides the POI’s faces, these images also contain the faces of 11 short-terms visitors who came to this floor during the experiments. We put different cameras in different rooms to examine the performance of AutoTune under camera heterogeneity. To further examine the resilience of AutoTune, we put cameras in adversarial positions. In kitchen, we deploy cameras with bad views near entrance so that they can only capture subjects above 1.7m. While in the meeting room with two entrances, only the primary entrance is equipped with cameras. Therefore in both rooms cameras constantly *mis-capture* face images of subjects. Tab. 1 summarizes this data collection.

CHN Site: We collect another dataset in a common room of a university in China. There are no long-term occupants in this site and all undergraduates can enter. Of the 37 people that appeared during the three week period, 12 subjects are selected as the POI, and their WiFi MAC address presence is continuously recorded by the sniffer. Other settings remain the same as the UK site. The challenge in this dataset lies in that the captured face images, both for POI and non-POI, are all of Asian people, while the initial face representation model is trained primarily on Caucasians. Details of the CHN dataset are given in Tab. 1. Observation noises are very common in this dataset.

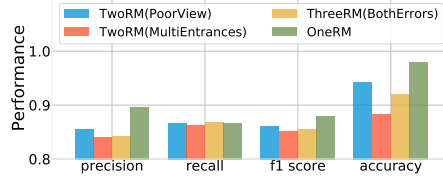
7.2.2 Overall Labeling Performance. We start our evaluation with one room only and compare our results with baselines. Fig. 5 shows the performance of label assignment, i.e., matching an identifier to a face image. For the office dataset, AutoTune outperforms the best competing approach (OA), by 0.13 in F_1 score and 7% in accuracy. The advantage of AutoTune is more obvious in the CommonRoom experiment where it beats the best competing approach (OA) by 0.34 in F_1 score and 10% in accuracy. As the only method that

Table 1: Key Metrics of Collected Dataset Sets.

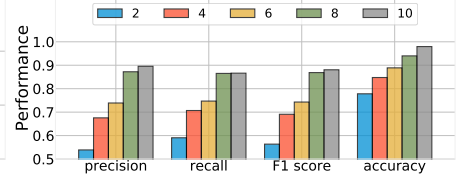
Site	NO. of Rooms	NO. of POI	NO. of non-POI	NO. of Images	NO. of Events	Event Duration	Average subjects/event	Camera(s) in rooms	Experiment Note
UK	3	24	11	15, 286	83	3h	9.14	Office: GoPro Hero 4 Kitchen: Pi Cam & Mi Cam Meeting: Pi Cam & Mi Cam	GoPro: 1080p, 90FPS PiCam: 720p, 90FPS MiCam: 720p, 15FPS
CHN	1	12	25	7, 495	102	2h	3.36	CommonRM: Gopro Hero 4	All Faces are from Asian



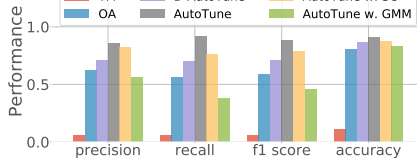
(a) Office, UK



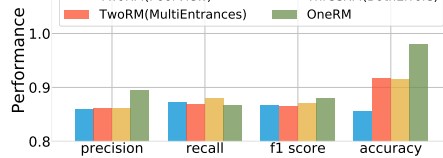
(a) Raspberry Pi Camera



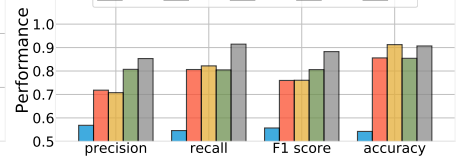
(a) UK



(b) CommonRoom, CHN



(b) Mi Smart Camera



(b) CHN

Figure 5: Overall Labeling Performance.

Figure 6: Performance of Scalability.

Figure 7: Impact of Number of Days.

uses the website images (one-shot learning) to label images rather than the device ID information, TM struggles in both experiments and is 9-fold worse than AutoTune. We observe that the website face images are dramatically different from the captured images in real-world, due to different shooting conditions and image quality. These results imply that, although the device observations via WiFi sniffing are noisy, when the amount of them is enough, they are more informative than the web-crawled face images.

Additionally, we note that adopting soft labels (see §4.2) can further improve the labeling performance. In terms of F_1 score, the full-suite AutoTune is around 15% and 22% better than D-AutoTune in the two experiments respectively. Similar improvements are witnessed in terms of the accuracy. The reason of the larger performance gap in the CommonRoom dataset is that there are more images of non-POI being captured in this experiment. In addition, the pre-trained FaceNet model does not generalize very well to Asian faces, which are the primary ethnicity in the CommonRoom.

Lastly, we found that the choice of algorithm for cross-event clustering affects the final label-association accuracy. The best performance is achieved with the default agglomerative clustering algorithm on both datasets. The second-best spectral clustering (SC) is slightly inferior to agglomerative clustering in the office dataset, though the gap between them gets large in the CommonRoom dataset. The Gaussian Mixture Model (GMM) is inferior to the other two clustering algorithms. This is because GMM is best suited to Mahalanobis-distance based clustering whereas the distance space defined in Eq. 4.1 is a non-flat space due to the introduced attendance information. Hence, its resultant clusters are very impure and give poor association performance.

7.2.3 Performance vs. Scalability. We further examine AutoTune with multiple rooms under different adversarial conditions. Together with the *UK office* data, we evaluate AutoTune when events are collected in three different locations and via heterogeneous cameras. In particular, a kitchen and a meeting room are included and each of them has two lower-fidelity cameras (Mi Home Camera and Raspberry Pi Camera). The setup of multi-location experiments is given in Tab. 1. As described in §7.2.1, there are many face mis-detections in the kitchen and meeting room due to the adversarial camera setups. Nevertheless, as shown in Fig. 6, AutoTune only suffers little performance drop (≤ 0.03) when erroneous camera observations are mixed with the single-office data. In terms of F_1 score, which is the most important metric, AutoTune can achieve comparable level with the office experiment, regardless of which camera is added. Moreover, AutoTune still maintains its good performance even in the most adversarial case, where both errors are mixed in (three-room experiment). The main reason is that by using more data, though noisy, extra validation constraints are also utilized by AutoTune and make itself robust to observation errors.

7.2.4 Performance vs. Lifespan. AutoTune exploits co-located device and face presence in different events to establish the cross-modality correlations. In this section, we investigate the impact of the collection span on the performance of labeling. Longer collection days give more events. We investigate its impact by feeding AutoTune with data collected in different number of days, and compare them with all days on two datasets respectively. Fig. 7 shows that AutoTune performs better with increasing number of days on both datasets. The gap of F_1 score between the case with all days

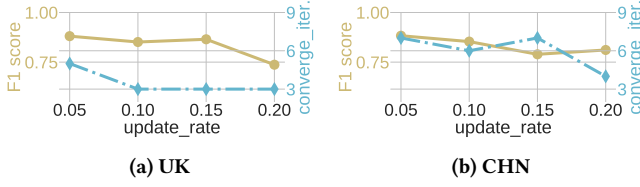


Figure 8: Impact of ID update rate β (introduced in §5.2). It can be seen that a high update rate causes the network to converge rapidly to potentially incorrect assignments.

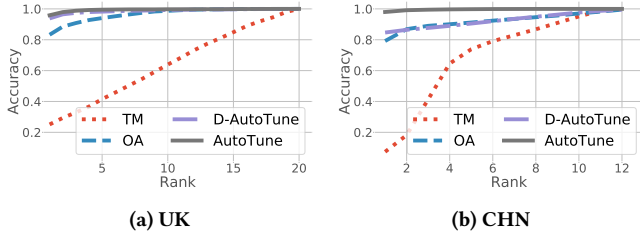


Figure 9: Online Identification Performance

(10 days) and case with the least amount of days (2 days) can be as large as > 0.32 on both datasets. As discussed in §4.2, the ID association needs sufficiently diverse events to create discriminative enough event vectors. Otherwise, there will be faces or devices with the same event vectors that hinders AutoTune’s ability to disambiguate their mutual association. However, we also observe that when we collect more than 8 days, the performance improvement of AutoTune becomes marginal.

7.2.5 Impact of ID update rate β . This section investigates the impact of ID update rate β introduced in §5.2. The ID update uses the fine-tuned face recognition model to update the device ID observations. A large ID update rate forces the device ID observations to quickly become consistent with the deep face model predictions. However, a large update rate also runs the risk of missing the optima. We vary the update rate β from 0.05 to 0.20 at a step length of 0.05. Fig. 8a demonstrates that AutoTune achieves the best performance on the UK dataset when the update rate is set to 0.05. The performance declines by 10% when the rate rises to 0.2. This is because the updated ID observations quickly become the same as the model predictions and the model predictions are not quite correct yet. When it comes to CHN dataset (see Fig. 8b), similar trend of F_1 score change can be seen. Although overall, the convergence becomes faster when the update increases, we observe that there is a fluctuation point at the update rate of 0.15, where AutoTune takes 7 iterations to converge. By inspecting the optimization process, we found that, under this parameter setting, AutoTune oscillated because the large update step makes it jump around in the vicinity of the optima but it is unable to approach it furthermore. In practice, we suggest users of AutoTune to select their update rate from a relatively safe region between 0.05 to 0.1.

7.3 Online Face Identification

7.3.1 Setup. After the cross-modality label association, an image database of POI is developed and an online face recognition system can be trained based on it. As in [26], we use the face representation model to extract face features and then use a classifier, say linear SVM in this work, to recognize faces. Particularly, as AutoTune and D-AutoTune are able to update the face representation model in their labeling process, the feature extractor for them are the updated model by themselves. For competing approaches that do not have an evolving model updates, we use the original Facenet model (in §6.2) as to extract face features and train the same classifier on the developed image-identity database by themselves. Without any overlapping with the images used for labeling, the test sets are collected in different days. In total, the test set contains 5,580 face images collected in the UK and 2,840 face images collected in China.

7.3.2 Performance. Fig. 9 compares the face identification results of both AutoTune and the competing approaches. Face identification is a multi-class classification task and aims to find an unknown person in a dataset of POI. As we can see, by using the face database developed by AutoTune, the identification accuracy quickly saturates and is able to have no errors within three guesses (rank-3) on both datasets. For the UK dataset where there are 20 POI, the rank-1 accuracy of AutoTune can be as high as 95.8% and outperforms the best competing approach (OA) by 12.5%. The advantage of AutoTune is more significant on the CHN dataset, and it surpasses the best competing approach OA by $\sim 19\%$ (98.0% vs 79.1%). In addition, although D-AutoTune’s performance is inferior to AutoTune, it is still more accurate than competing approaches, especially on the UK dataset. We note that these results are consistent with face labeling results in §7.2.2. Overall, face identification by AutoTune are highly accurate, considering that AutoTune is only supervised by the weak and noisy device presence information.

7.4 Sensitivity Analysis

To further study the performance of AutoTune under different noise conditions, we also conduct extensive sensitivity analysis via simulation, considering three types of noise that are common in real-world settings:

False-alarm Faces is the case that the number of detected faces of POI is *greater than* the number of distinct sniffed MAC addresses.

False-alarm Devices is the case that the number of detected faces of POI is *smaller than* the number of distinct sniffed MAC addresses.

Non-POI Disturbance is the case that there are detected faces that belong to a non-POI, whose MAC address are not in our database i.e. they should be discarded as an outlier.

7.4.1 Data Synthesis. Although a number of public databases of face images are available, unfortunately, none of them come with device presence observations, i.e., sniffed MAC addresses. As an alternative, we develop a synthetic dataset on top of the widely used PIE face database⁹. In total 40 subjects are randomly selected, in which 30 of them are POI and the rest are mixed in to make the dataset realistic. We then assign to these “events” noisy identity observations to simulate device presence information. Based on different types of noise, different synthetic datasets are generated on which AutoTune is examined. On average, each subject in our

⁹<http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html>

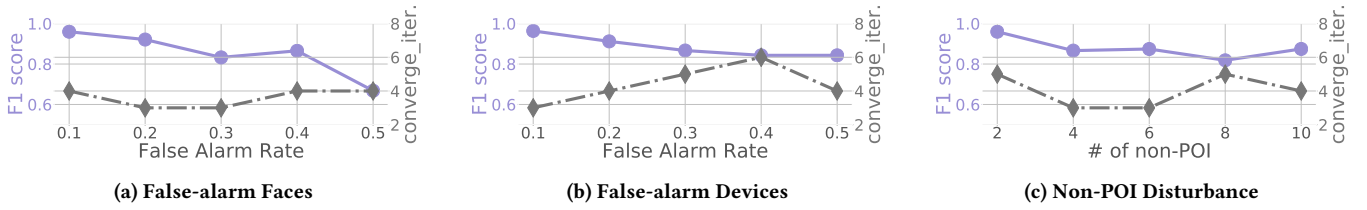


Figure 10: Results on simulation data showing the impact of sources of noise

simulation has 168 images after data augmentation [12]. As there is no actual device presence, we simulate this by first randomly placing the face images into different “events”, in which multiple subjects “attend”. As the number of images of subjects might be skewed in events, we adopt the F_1 score as the metric to evaluate the AutoTune’s performance.

7.4.2 False-alarm Faces. WiFi sniffing is an effective approach to detect the presence of users’ mobile devices, however, such detection is not guaranteed to perfectly match the user’s presence. For instance, a device could be forgotten at home, be out of battery or simply the WiFi function might be turned off. In the simulation, we vary the error rate of such false alarm faces from 0.1 to 0.5. Error rate at 0.1 means that on average, 10% of the detected faces in each event are false detections. Fig. 10a shows the F_1 score and convergence iterations of AutoTune under different levels of such noises. As we can see AutoTune tolerates false-alarm faces well and is able to keep the F_1 score above 0.83 when the false alarm rate is below 0.4, though it degrades to 0.67 when the rate rises up to 0.5. However, we found such case, i.e., on average half of WiFi MAC addresses are missed in all meetings, is rare in the real-world. Finally, we found that false-alarm faces do not affect the convergence and AutoTune quickly converges within 4 iterations in all the cases.

7.4.3 False-alarm Devices. Though surveillance cameras are becoming increasingly ubiquitous, there are cases where the subjects are not captured by the cameras e.g. duct occlusions. This becomes an instance of device false alarm, if her device MAC address is still sniffed. We vary the rate of such false alarm devices from 0.1 to 0.5, where 0.1 means that on average, 10% of the detected devices in each event are false detections. Fig. 10b shows that although the F_1 score of AutoTune decreases, it degrades slowly and stops at 0.84 after the false-alarm rate becomes 0.5. As the injected noise becomes stronger, AutoTune needs more iterations to converge. However, the largest convergence iteration is still below 7 (rate at 0.4). Overall, AutoTune is very robust to such type of noises.

7.4.4 Non-POI Disturbance. Non-POI disturbance happens when subjects without registered MAC addresses are captured by the camera. We found such noise dominates all the three types of errors. We vary the number of non-POI from 2 to 10 and the probability of each non-POI’s presence in an event is set to 0.1. Fig. 10c shows that AutoTune does not suffer much from mild disturbance (2 non-POI), and the F_1 score drops slowly to 0.87 with larger disturbance (4 non-POI). In addition in all cases AutoTune quickly converges within 5 iterations.

8 DISCUSSION AND FUTURE WORK

This section discusses some important issues related to AutoTune.

Overheads: Compared with conventional face recognition methods, AutoTune incurs overheads due to FaceNet fine-tuning. In our experiment, fine-tuning is realized on one NVIDIA K80 GPU. One fine-tuning action takes around 1.2 hours and 0.5 hours for the UK dataset and CHN dataset respectively. As we discuss in §7.2, AutoTune is able to converge within 5 iterations for most of the time, depending on the hyper-parameter setting. Therefore, fine-tuning overheads can be controlled in 7 hours for the UK dataset and 2.5 hours for the CHN dataset. Compared to the FaceNet pre-training that takes days and requires more GPUs, the fine-tuning costs of AutoTune is much cheaper. In future work, we will look into an online version of AutoTune, which can incrementally fine-tune the network on the fly when the data is streaming in.

Privacy: In practice, AutoTune requires face images and device ID of users to operate, which may have certain impacts on user privacy. For example, a user could be identified without explicit consent in a new environment, if the owner has the access to the face image of this user. In this work, we do not explicitly study the attack model in this context, we note that potential privacy concerns are worth exploring in future work.

9 CONCLUSION

In this work, we described AutoTune, a novel pipeline to simultaneously label face images in the wild and adapt a pre-trained deep neural network to recognize the faces of users in new environments. A key insight that motivates it is that enrolment effort of face labelling is unnecessary if a building owner has access to a wireless identifier, e.g., through a smart-phone’s MAC address. By learning and refining the noisy and weak association between a user’s smart-phone and facial images, AutoTune can fine-tune a deep neural network to tailor it to the environment, users, and conditions of a particular camera or set of cameras. Particularly, a novel soft-association technique is proposed to limit the impact of erroneous decisions taken early on in the training process from corrupting the clusters. Extensive experiment results demonstrate the ability of AutoTune to design an environment-specific, continually evolving facial recognition system with entirely no user effort even if the face and WiFi observations are very noisy.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the EPSRC through grant EP/M019918/1 and Oxford-Google DeepMind Graduate Scholarship.

REFERENCES

- [1] Alexandre Alahi, Albert Haque, and Li Fei-Fei. 2015. RGB-W: When vision meets wireless. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 3289–3297.
- [2] Doug Beeferman and Adam Berger. 2000. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 407–416.
- [3] Lacey Best-Rowden, Hu Han, Charles Otto, Brendan F Klare, and Anil K Jain. 2014. Unconstrained face recognition: Identifying a person of interest from a media collection. *IEEE Transactions on Information Forensics and Security* 9, 12 (2014), 2144–2157.
- [4] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. 2018. VGGFace2: A dataset for recognising faces across pose and age. In *IEEE Conference on Automatic Face and Gesture Recognition*.
- [5] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. 2016. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1335–1344.
- [6] Chris Fraley and Adrian E Raftery. 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The computer journal* 41, 8 (1998), 578–588.
- [7] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. 2013. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*. 2121–2129.
- [8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
- [9] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. 2007. *Labeled faces in the wild: A database for studying face recognition in unconstrained environments*. Technical Report. Technical Report 07-49, University of Massachusetts, Amherst.
- [10] Roy Jonker and Ton Volgenant. 1986. Improving the Hungarian assignment algorithm. *Operations Research Letters* 5, 4 (1986), 171–175.
- [11] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3128–3137.
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
- [13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.
- [14] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2017. Sphreface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [15] Weiyang Liu, Yan-Ming Zhang, Xingguo Li, Zhiding Yu, Bo Dai, Tuo Zhao, and Le Song. 2017. Deep Hyperspherical Learning. In *Advances in Neural Information Processing Systems*. 3953–3963.
- [16] Chris Xiaoxuan Lu, Xuan Kan, Stefano Rosa, Bowen Du, Hongkai Wen, Andrew Markham, and Niki Trigoni. 2017. Towards Self-supervised Face Labeling via Cross-modality Association. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*.
- [17] Chris Xiaoxuan Lu, Hongkai Wen, Sen Wang, Andrew Markham, and Niki Trigoni. 2017. SCAN: learning speaker identity from noisy sensor data. In *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 67–78.
- [18] Xiaoxuan Lu, Hongkai Wen, Han Zou, Hao Jiang, Lihua Xie, and Niki Trigoni. 2016. Robust occupancy inference with commodity WiFi. In *IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*.
- [19] Xiaoxuan Lu, Han Zou, Hongming Zhou, Lihua Xie, and Guang-Bin Huang. 2016. Robust extreme learning machine with its application to indoor positioning. *IEEE transactions on cybernetics* 46, 1 (2016), 194–205.
- [20] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. 2018. Seeing Voices and Hearing Faces: Cross-modal biometric matching. *arXiv preprint arXiv:1804.00326* (2018).
- [21] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. 2016. Ambient sound provides supervision for visual learning. In *European Conference on Computer Vision*. Springer, 801–816.
- [22] Savvas Papaioannou, Hongkai Wen, Zhuoling Xiao, Andrew Markham, and Niki Trigoni. 2015. Accurate positioning via cross-modality training. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 239–251.
- [23] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. 2015. Deep Face Recognition.. In *BMVC*, Vol. 1. 6.
- [24] David Martin Powers. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2011).
- [25] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596* (2014).
- [26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.
- [27] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. 2014. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*. 1988–1996.
- [28] Yi Sun, Xiaogang Wang, and Xiaoou Tang. 2014. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1891–1898.
- [29] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning.. In *AAAI*.
- [30] Yaniv Taigman, Ming Yang, Marc Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1701–1708.
- [31] Xiaoyang Tan, Songcan Chen, Zhi-Hua Zhou, and Fuyan Zhang. 2006. Face recognition from a single image per person: A survey. *Pattern recognition* 39, 9 (2006), 1725–1745.
- [32] Jin Teng, Boying Zhang, Junda Zhu, Xinfeng Li, Dong Xuan, and Yuan F Zheng. 2014. EV-Loc: integrating electronic and visual signals for accurate localization. *IEEE/ACM Transactions on Networking (TON)* 22, 4 (2014), 1285–1296.
- [33] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2014. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729* (2014).
- [34] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 3156–3164.
- [35] Stefan Wager, Sida Wang, and Percy S Liang. 2013. Dropout training as adaptive regularization. In *Advances in neural information processing systems*. 351–359.
- [36] Junjue Wang, Brandon Amos, Anupam Das, Padmanabhan Pillai, Norman Sadeh, and Mahadev Satyanarayanan. 2018. Enabling Live Video Analytics with a Scalable and Privacy-Aware Framework. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 3s (2018), 64.
- [37] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A discriminative feature learning approach for deep face recognition. In *ECCV*.
- [38] Xuanli Lisa Xie and Gerardo Beni. 1991. A validity measure for fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence* 13, 8 (1991), 841–847.
- [39] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503.
- [40] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. 2003. Face recognition: A literature survey. *ACM computing surveys (CSUR)* 35, 4 (2003), 399–458.
- [41] C Lawrence Zitnick, Devi Parikh, and Lucy Vanderwende. 2013. Learning the visual interpretation of sentences. In *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 1681–1688.